



**QUEEN'S  
UNIVERSITY  
BELFAST**

## A biologically inspired appearance model for robust visual tracking

Zhang, S., Lan, X., Yao, H., Zhou, H., Tao, D., & Li, X. (2017). A biologically inspired appearance model for robust visual tracking. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2357-2370. <https://doi.org/10.1109/TNNLS.2016.2586194>

**Published in:**  
IEEE Transactions on Neural Networks and Learning Systems

**Document Version:**  
Peer reviewed version

**Queen's University Belfast - Research Portal:**  
[Link to publication record in Queen's University Belfast Research Portal](#)

### **Publisher rights**

© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

### **General rights**

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

### **Take down policy**

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact [openaccess@qub.ac.uk](mailto:openaccess@qub.ac.uk).

# A Biologically Inspired Appearance Model for Robust Visual Tracking

Shengping Zhang, *Member, IEEE*, Xiangyuan Lan, Hongxun Yao, *Member, IEEE*, Huiyu Zhou, Dacheng Tao, *Fellow, IEEE*, Xuelong Li, *Fellow, IEEE*

**Abstract**—In this work, we propose a biologically inspired appearance model for robust visual tracking. Motivated in part by the success of the hierarchical organization of the primary visual cortex (area V1), we establish an architecture consisting of five layers: whitening, rectification, normalization, coding and pooling. The first three layers stem from the models developed for object recognition. In this paper, our attention focuses on the coding and pooling layers. In particular, we use a discriminative sparse coding method in the coding layer along with spatial pyramid representation in the pooling layer, which makes it easier to distinguish the target to be tracked from its background in the presence of appearance variations. An extensive experimental study shows that the proposed method has higher tracking accuracy than several state-of-the-art trackers.

**Index Terms**—Visual tracking, Appearance modeling, Biologically inspired, Sparse coding

## I. INTRODUCTION

VISUAL tracking is a task that continuously infers the state of a specific target from an image sequence; it is a specific task of computer vision which has attracted increasing interest in recent years [1]–[15]. Given a target template, a visual tracking process usually consists of four stages: 1) Candidate sampling, whereas we sample a set of target candidates from the current image frame. 2) Appearance modeling, where we describe the target template and a number of candidates for correspondence using an appearance representation method. 3) Target searching, which allows us to find which candidate has the best similarity to the target template. 4) Template updating, where we update the appearance of the target template in order to adapt to the variations of the target appearance over time.

In these four stages, appearance modeling attracts large attention in the last few decades [3], [16]–[29]. A good appearance model should not only be used to distinguish the target from its background but also should be robust against

appearance variations due to pose changes, illumination and occlusions. Although many appearance models have been reported in the literature, they still do not have sufficient capabilities to handle these challenges. One of the main reasons is that most of these models are based on handcrafted features [30]–[32]. The computation procedures of these features are fixed regardless of input data and therefore cannot be adapted to different tracking scenarios. Several recent models resort to machine learning methods and devise suitable training criteria and optimization methods to learn features from data [33]–[35]. In spite of their high adaptive capability, these methods still somehow depend on prior knowledge, which does not easily transfer to other applications.

While robust visual tracking remains very challenging to computers, it is effortless for humans to accomplish this task. The progress made in understanding the brain mechanisms of visual information processing, especially the primary visual cortex (area V1), enable us to develop effective appearance models to handle the tracking problems. In this paper, we focus on developing a biologically inspired appearance model for robust visual tracking. The motivation is twofold:

- *Motivation (i)*: psychophysical evidence has shown that V1 can process visual information using a hierarchical structure [36], [37], which alternates between simple and complex cells. Riesenhuber and Poggio [38] proposed a quantitative model to simulate such a hierarchy, where the responses of simple cells were produced using local filters whilst the invariance properties of complex cells were created using a max pooling process over the neighboring positions.
- *Motivation (ii)*: Olshausen and Field [39] proposed an unsupervised learning method, called sparse coding, to represent a natural image as a linear combination of a set of basis functions. Assume the coefficients are sparse and statistically independent. People have observed that these basis functions captured the underlying image structures (lines and edges). More importantly, the basis functions have similar response properties to those of the simple cells in area V1.

Motivation (i) suggests that we can perform appearance modeling for visual tracking by simulating the hierarchical structure of area V1. In the literature, several appearance models with this motivation have been reported for object recognition [40]–[42], which used one or two stages of V1-like feature representation. The responses of simple cells at the bottom of the hierarchy was generated using Gabor filters.

S. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai, P. R. China. E-mail: s.zhang@hit.edu.cn.

X. Lan is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, P. R. China. E-mail: xylan@comp.hkbu.edu.hk.

H. Yao is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P. R. China. E-mail: h.yao@hit.edu.cn.

H. Zhou is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University of Belfast, Belfast, BT3 9DT, United Kingdom, E-mail: h.zhou@qub.ac.uk.

D. Tao is with the Center for Quantum Computation and Intelligence Systems, University of Technology, Sydney, Australia. E-mail: dacheng.tao@uts.edu.au.

X. Li is with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China. Email: xuelong\_li@opt.ac.cn

Corresponding author: Shengping Zhang

At the top of the hierarchy, complex cells were simulated by performing max pooling over the local neighboring responses. Their experimental results show that these models outperform the other state-of-the-art object recognition systems on several standard datasets [40], [41].

Although these models are biologically inspired, simple cells were only simulated by handcrafted Gabor filters, resulting in a relatively poor evolutionary capability in their systems. In fact, neurons in the visual cortex should be adapted to different statistical properties of the visual signals to which they are exposed [43], [44]. Therefore, a more reasonable simulation of simple cells in visual cortex is to use learning based methods to capture the statistical properties of the visual signals. Sparse coding shown in the motivation (ii) is a model that meets this requirement. Therefore, we here propose to use sparse coding to model the responses of simple cells in V1. Furthermore, considering the importance of the discriminative ability of an appearance representation for visual tracking, we therefore use discriminative sparse coding and spatial pyramid pooling to make the representation more capable of distinguishing the target from its background.

The architecture of the proposed biologically inspired appearance model is shown in Fig. 1, which consists of five layers. The whitening, rectification and normalization layers stem from object recognition models and perform very well in various applications [41], [42]. However, in visual tracking applications, we need to maintain the discriminative capability of the appearance model employed in the tracker all the time, which is more challenging than the case of object recognition, and therefore, our attention in this paper focuses on the coding and pooling layers. In particular, in the coding layer, we will adopt a discriminative sparse coding technique to represent local patches densely sampled from the input image. This makes it easy to distinguish between the target and the background patches. In the pooling layer, average pooling with spatial pyramid representation will be used to aggregate the local codes into a global representation of the input image. The multiple scale pyramid structure can be used to preserve the spatial relationship of the local codes, and hence increase the global discriminative capability of the final feature representation with strong robustness against local appearance variations.

The rest of this paper is organized as follows. In Section II, we review the work closely related to our proposed approach. Section III gives a detailed description of the proposed biologically inspired appearance model. The tracking algorithm based on the proposed appearance model is introduced in Section IV. Experimental results are reported and analyzed in section V. We conclude the paper in Section VI.

## II. RELATED WORK

In this section, we review the biologically inspired appearance models used for object recognition, state-of-the-art tracking approaches and corresponding benchmarks.

### A. Biologically inspired models for object recognition

Based on the findings in human psychophysics and electrophysiology studies [36], [37], [45], Riesenhuber and Poggio

[38] proposed a quantitative model to simulate the hierarchical structure in human vision for object recognition, where the response of the simple cells was obtained using the second derivative of Gaussian filters, and the invariance properties of the complex cells were obtained using a max pooling operator over the neighboring positions. Serre et al. [40] proposed the HMAX model which uses Gabor filters to replace the second derivative of Gaussian filters used in [38]. In addition, they also proposed to use a two-stage V1 like feature representation. In the second stage of the system, the filters were built by randomly sampling the outputs of the first stage. The max pooling of the filters' response over the entire image forms the final feature representation. Pinto et al. [41] simplified the HMAX model and only used the first stage of the HMAX model with additional preprocessing and normalization operators. In addition, they used a sum operator instead of a max operator in the pooling stage. The experiments against several object recognition datasets demonstrate that their system achieved better performance than the other state-of-the-art systems. Yang et al. [46] improved the bag-of-features (BoF) model [47] by using sparse coding to replace the K-means in the vector quantization step. Although they do not claim their method is biologically inspired, their feature extraction method is very similar to that of [41]. Other researchers such as [42], [48], [49] conducted extensive experiments in object recognition in order to compare the performance of different hierarchical architectures. The evaluation results indicate that a single layer model with coding, pooling and suitable preprocessing and normalization operators can achieve the expected recognition performance.

### B. State-of-the-art tracking approaches and corresponding benchmarks

Visual tracking has attracted increasing interests in recent years. A large number of novel tracking approaches, large benchmark datasets and experimental comparison were reported [1], [6], [50]–[54]. In [50], a kernelized structured output support vector machine was learned online to achieve adaptive tracking. Real-time tracking was achieved by introducing a budgeting mechanism which prevents the unbounded growth of the number of support vectors which would otherwise occur during tracking. Very recently, deep learning has been successfully applied to different vision tasks. Inspired by recent advances in deep learning architectures, Wang et al. proposed a deep compact image representation for visual tracking [51], which first trains a stacked denoising autoencoder offline to learn generic image features that are more robust against variations and then transfers the offline training to an online tracking process. In [55], Chen et al. used cells to extract local appearance, and constructed complex cells to integrate the information from cells to provide diverse and important object cues for visual tracking. With different spatial arrangements of cells, complex cells are of various contextual information at multiple scales, which is important to improve the tracking accuracy. Although two biological terms such as “cells” and “complex cells” were used in their paper, there is little biologically inspiration. In [6], the authors proposed a

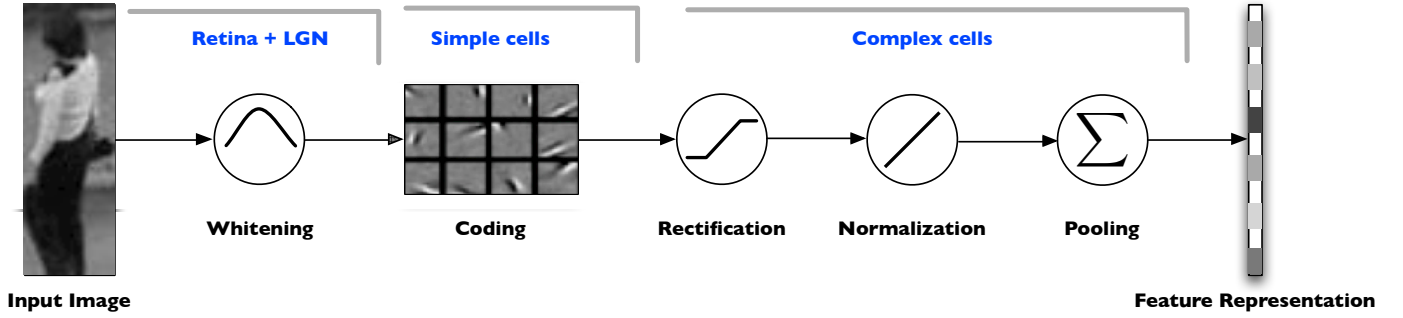


Fig. 1. The architecture of the proposed biologically inspired appearance modeling framework (LGN = Lateral Geniculate Nucleus).

new Kernelized Correlation Filter (KCF) for visual tracking, which can achieve fast and accurate tracking. To evaluate the progress in visual tracking, there are also several benchmark databases and experimental surveys on recent tracking methods. In [1], a benchmark with 50 test sequences was built for evaluating state-of-the-art tracking methods, which is then further extended in [54] to have 100 test sequences. In [53], a set of nineteen trackers were evaluated on 315 video fragments, which provides objective insight into the strengths and weaknesses of these trackers.

In the literature, several appearance models based on sparse coding have been proposed for visual tracking, which usually consist of two steps: local coding and global pooling. After the input image is divided into a set of image patches, each patch is sparsely coded by a linear combination of a set of basis functions. The combination coefficients are used as codes to describe the image patch. The final appearance representation of the input image is obtained by pooling the codes of all the patches to form a histogram-like feature vector. Zhang et al. [56] proposed to code image patches using independent component analysis (ICA) basis functions, which were learned from a set of randomly sampled image patches. Average pooling was used to obtain the final appearance representation. In [19], each patch was coded by the basis functions of sparse coding, which were learned using the sampled patches out of the target image. Similar to [56], average pooling was used to construct the final feature representation. Wang et al. [57] combined the sampled patches from the target image and the identity vectors as basis functions to code each patch. All the codes were concatenated together to form the final feature representation. Zhong et al. [58] learned the basis functions of sparse coding from the patches sampled from the target image via K-means. The pooling step is similar to the one used in [57]. Jia et al. [20] maintained multiple target templates. Local patches sampled from each template were collected and formed as basis functions to code patches sampled from the input image with the same spatial layout. They used an alignment pooling operator to preserve the structural relationship between local codes.

Although these appearance models, based on sparse coding, have shown promising results in some applications, they also accompany some inevitable shortcomings: 1) the basis functions of sparse coding are obtained by either learning

from images [56] or sampling the patches from the tracking data [20]. The basis functions are so generative that the resulting sparse codes from the target are not discriminative enough to distinguish the target from its background. 2) The sparse codes are pooled by either summing all the coefficient vectors over the entire image which loses the spatial relationship [19], [56], or concatenating all the coefficient vectors which preserves the spatial relationship but also violating the local invariance of the appearance representation [57]. 3) Supported by the biological studies [59], [60] and empirical evaluation [41], [42], non-linear operators such as whitening, rectification and normalization operators can also enhance the representation capability. However, the existing appearance models based on sparse coding only contain the basic coding and pooling operators and therefore ignore the role of the non-linear operators.

### III. PROPOSED BIOLOGICALLY INSPIRED APPEARANCE MODEL

The visual information received by simple cells in V1 is just part of the information contained in the whole retina image, which is determined by the size of the receptive field of simple cells. To simulate this mechanism, first of all, a set of  $n$  patches are densely sampled from an input image  $\mathbf{I}$ . Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$  be the sampled patch set, where  $\mathbf{x}_i \in \mathbb{R}^d$  refer to the stacked pixel intensities from the  $i$ -th patch. To reduce the impact of noise, each patch is first normalized by subtracting its mean and then divided by its standard deviation.

The proposed appearance modeling method is used to find a feature mapping function  $\mathbf{y} = \mathcal{F}(\mathbf{X})$ , which takes in the patch set  $\mathbf{X}$  and then outputs a feature vector  $\mathbf{y}$  that is capable of representing the appearance of the image  $\mathbf{I}$ . Under the hierarchical framework as shown in Fig. 1, let  $\mathcal{W}$ ,  $\mathcal{C}$ ,  $\mathcal{R}$ ,  $\mathcal{N}$  and  $\mathcal{P}$  be the feature mappings of the whitening, coding, rectification, normalization and pooling layers, respectively. The feature mapping function  $\mathcal{F}$  can be obtained by sequencing all the layers as  $\mathcal{F} = \mathcal{P} \circ \mathcal{N} \circ \mathcal{R} \circ \mathcal{C} \circ \mathcal{W}$ . Algorithm 1 summarised the steps of our appearance modelling method. In the following sub-sections, we will introduce how to implement these feature mappings in details.



**Algorithm 1:** The proposed appearance modelling method.

---

**Input** : Input image  $\mathbf{I}$ , parameters  $n$  and  $\lambda$   
**Output:** Appearance representation  $\mathbf{y}$

- 1 Sample  $n$  patches  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  from the input image  $\mathbf{I}$ ;
- 2 Whiten each patch  $\mathbf{x}_i$  in  $\mathbf{X}$  to have the whitened patch  $\mathbf{w}_i$ ;
- 3 Code each whitened patch  $\mathbf{w}_i$  using the learned discriminative dictionary to get the responses  $\mathbf{c}_i$ ;
- 4 Rectify the responses of a whitened patch to obtain the rectified responses  $\mathbf{r}_i$ ;
- 5 Normalize the rectified responses of a whitened patch to gain the normalised responses  $\mathbf{n}_i$ ;
- 6 Integrate the normalised responses in a pyramid division to obtain the feature representation  
 $\mathbf{y} = \{\mathbf{y}_{s,v,h} | s = 0, 1, 2, v = 1, 2, \dots, 2^s, h = 1, 2, \dots, 2^s\}$

---

*A. Whitening layer*

A natural scene observed by the human eye is transmitted to the retina. Since the adjacent pixels of the image tend to be highly correlated in intensity, it is wasteful for the retina to transmit every pixel separately to the brain for further processing. Instead of doing this, the retina performs a decorrelation operation via retinal neurons, which can be implemented using the ZCA whitening operator [61]. To whiten each patch in  $\mathbf{X}$ , the covariance matrix is firstly computed as  $\mathbf{X}\mathbf{X}^\top$ , where  $\top$  denotes matrix transpose. The eigen-value decomposition of the covariance matrix is  $\mathbf{E}\mathbf{D}\mathbf{E}^\top$  where the orthogonal matrix  $\mathbf{E}$  consists of eigenvectors and the diagonal matrix  $\mathbf{D}$  consists of eigenvalues. Then the whitening operator can be achieved as follows:

$$\mathbf{w}_i = \mathcal{W}(\mathbf{x}_i) = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^\top\mathbf{x}_i \quad (1)$$

After the whitening, all the dimensions of the image patches are uncorrelated and their variances are equal. For computation, the whitening operator is also useful to speed up the convergence of the coefficient inference in the subsequent coding layer.

*B. Coding layer*

The goal of the coding layer is to extract local features from each image patch. In contrast to other handcrafted features, in this work, we consider using sparse coding to learn sparse features for each image patch. The motivation of using sparse coding to extract features is twofold: 1) the sparse features are capable of capturing the underlying structural properties of the images due to small reconstruction errors and the sparsity constraints in the dictionary learning. 2) Studies in neuroscience clearly reveal that the human brain codes visual information in a sparse manner.

Given a dictionary with  $K$  basis functions  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_K] \in \mathbb{R}^{d \times K}$ , the coding responses of a whitened patch  $\mathbf{w}_i$  can be computed as

$$\mathbf{c}_i = \mathcal{C}(\mathbf{w}_i) = \arg \min_{\mathbf{c}} \|\mathbf{w}_i - \mathbf{V}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1 \quad (2)$$

where  $\lambda$  is a regularization parameter that controls the importance of the sparsity constraint of the responses to the reconstruction error. Similar to the design of the handcrafted filters [38], [40], a critical step in the coding layer is to learn the dictionary  $\mathbf{V}$  from a set of image patches  $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{\tilde{n}}] \in \mathbb{R}^{d \times \tilde{n}}$  which are sampled from a set of training images. After the whitening, these image patches supply training samples  $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_1, \tilde{\mathbf{w}}_2, \dots, \tilde{\mathbf{w}}_{\tilde{n}}] \in \mathbb{R}^{d \times \tilde{n}}$  for the dictionary learning, where  $\tilde{\mathbf{w}}_i = \mathcal{W}(\tilde{\mathbf{x}}_i)$ . The traditional sparse coding scheme learns the dictionary by solving the following equations:

$$\arg \min_{\mathbf{V}, \mathbf{c}_i} \sum_{i=1}^{\tilde{n}} \|\tilde{\mathbf{w}}_i - \mathbf{V}\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_1 \quad (3)$$

The objective function above only contains the reconstruction error and sparse coding responses, which is essentially required for learning a sparse but approximate representation. However, such a representation is not good enough for those applications, which require high discriminative ability, e.g., the visual tracking task studied in this work. To make the coding responses easier to discriminate the target from its background, in this work, we utilize a discriminative dictionary learning method [62], which learns two dictionaries for the target and the background, respectively. We call them as the target dictionary and the background dictionary accordingly in the later sections of this paper. The target dictionary represents the patches sampled from the target image with the minimal reconstruction error, while representing the patches sampled from the background image with the maximal reconstruction error. Similarly, the background dictionary represents the background patches with the minimal reconstruction error, while representing the target patches with the maximal reconstruction error. The objective of using this dictionary learning method is to produce discriminative codes in the coding layer when we use the dictionary consisting of both the target and background dictionaries.

Let  $\tilde{\mathbf{W}}_f = [\tilde{\mathbf{w}}_{f,1}, \tilde{\mathbf{w}}_{f,2}, \dots, \tilde{\mathbf{w}}_{f,n_f}] \in \mathbb{R}^{d \times n_f}$  and  $\tilde{\mathbf{W}}_b = [\tilde{\mathbf{w}}_{b,1}, \tilde{\mathbf{w}}_{b,2}, \dots, \tilde{\mathbf{w}}_{b,n_b}] \in \mathbb{R}^{d \times n_b}$  be the training patches sampled from the target and the background images, respectively. Let  $\mathbf{V}_f \in \mathbb{R}^{d \times K_f}$  and  $\mathbf{V}_b \in \mathbb{R}^{d \times K_b}$  be the dictionary pair learned from the target and the background patches, respectively. The reconstruction error of coding a sample  $\tilde{\mathbf{w}}$  using the dictionary  $\mathbf{V}$  is computed as  $\mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}) = \|\tilde{\mathbf{w}} - \mathbf{V}\mathcal{C}(\tilde{\mathbf{w}})\|_2^2$ . To learn and create the dictionary pair  $\mathbf{V}_f$  and  $\mathbf{V}_b$ , we define the following discriminative cost function:

$$\mathcal{L}(\tilde{\mathbf{w}}, \mathbf{V}_f, \mathbf{V}_b) = \log(1 + e^{\lambda(\mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_f) - \mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_b))}) \quad (4)$$

where  $\lambda > 0$  is a parameter to be determined.  $\mathcal{L}(\tilde{\mathbf{w}}, \mathbf{V}_f, \mathbf{V}_b)$  is close to zero when  $\mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_f) < \mathcal{E}(\tilde{\mathbf{w}}, \mathbf{V}_b)$ . The discriminative dictionary pair  $\mathbf{V}_f$  and  $\mathbf{V}_b$  can be established by optimizing this cost function:

$$\begin{aligned} \min_{\mathbf{V}_f, \mathbf{V}_b} = & \frac{1}{n_f} \sum_{i=1}^{n_f} \mathcal{L}(\tilde{\mathbf{w}}_{f,i}, \mathbf{V}_f, \mathbf{V}_b) + \gamma \mathcal{E}(\tilde{\mathbf{w}}_{f,i}, \mathbf{V}_f) \\ & + \frac{1}{n_b} \sum_{i=1}^{n_b} \mathcal{L}(\tilde{\mathbf{w}}_{b,i}, \mathbf{V}_b, \mathbf{V}_f) + \gamma \mathcal{E}(\tilde{\mathbf{w}}_{b,i}, \mathbf{V}_b) \end{aligned} \quad (5)$$

where  $\gamma > 0$  is a parameter to be determined, and  $n_b$  and  $n_f$  are two numbers of the investigated patches. Given the randomly initialized  $\mathbf{V}_f$  and  $\mathbf{V}_b$ , this optimization problem can be solved by a standard K-SVD technique [63] with typically 5 iterations, consisting of two steps: 1) Coding—compute codes for each patch in  $\tilde{\mathbf{W}}_f$  (or  $\tilde{\mathbf{W}}_b$ ) using dictionary  $\mathbf{V}_f$  (or  $\mathbf{V}_b$ ). 2) Updating—update each column of  $\mathbf{V}_f$  or  $\mathbf{V}_b$  to reduce the residual of the cost function. After learning and updating the dictionaries  $\mathbf{V}_f$  and  $\mathbf{V}_b$ , we then create a combination form as  $\mathbf{V} = [\mathbf{V}_f, \mathbf{V}_b] \in \mathbb{R}^{d \times K}$  where  $K = K_f + K_b$ .

After obtaining the dictionary, we have the coding outputs of the whitened patches  $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n\}$  as  $\{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n\}$ , where  $\mathbf{c}_i = \mathcal{C}(\mathbf{w}_i)$  is computed using Eq. (2) with the learned dictionary.

### C. Rectification layer

Neuroscience studies [59], [64] indicate that simple cells in visual cortex are rarely in their maximum saturation regime and suggest that their response can be approximated by a rectification operator:

$$\mathbf{r}_i = \mathcal{R}(\mathbf{c}_i) = \max(\mathbf{0}, \mathbf{c}_i) \quad (6)$$

where  $\max(\cdot, \cdot)$  operates on two vectors and returns the maximal value of each dimension. The rectification layer makes the response to the opposite of an excitatory pattern to be 0 (*i.e.* no response). On the other hand, the outputs of the coding layer, the coefficients, still possess some small but non-zero values. The rectification operator enables the codes much sparser than an usual case, which has been proved to achieve better object recognition performance [42].

### D. Normalization Layer

Empirical evidence shows that the rectified coefficients, corresponding to the basis functions at the neighboring spatial positions, orientations and scales, are highly correlated. An operator of removing these dependencies is divisive normalization [60], [65], which represents each coefficient by a weighted combination of its rectified neighbors. Let  $\mathbf{r}_i$  and  $\mathbf{r}_i^1, \mathbf{r}_i^2, \dots, \mathbf{r}_i^P$  be the rectification layer's outputs of the image patch  $\mathbf{x}_i$  and its  $P$  neighboring patches, respectively, the divisive normalization for the  $k$ -th dimension can be implemented by

$$\mathbf{n}_i(k) = \mathcal{N}(\mathbf{r}_i(k)) = \frac{\mathbf{r}_i(k)}{\sum_{p=1}^P w_p \mathbf{r}_i^p(k) + \delta} \quad (7)$$

where  $\mathbf{r}(k)$  denotes the  $k$ -th element of the vector  $\mathbf{r}$ ,  $\mathbf{w} = [w_1, w_2, \dots, w_P]$  are the weights of the neighboring patches and  $\delta$  is a constant. The parameters  $\mathbf{w}$  and  $\delta$  can be learned by collecting a set of patches and then minimizing the prediction error. To simplify the model, we here use equal weights  $w_p = \frac{1}{P}$  and an infinitesimal value  $\delta$ .

### E. Pooling layer

The purpose of appearance modeling is to find a global feature representation to describe the target to be tracked. After

obtaining the rectified and normalized coding responses of the local patches sampled from the input image, we combine these local codes to form a global representation. This can be implemented by a pooling operator, which employs the statistics of the local sparse codes. Biological studies have shown that the complex cells in area V1 also use a pooling operator to make the final representation invariant to small distortions. Some biological models recruited sum pooling, *e.g.* [37], which computed the sum of responses to a specific stimuli over a neighboring space. Recently, people proposed to use max pooling instead of sum pooling [40], [66], which used a max operator to replace the sum operator in the sum pooling. Although max pooling can produce a more invariant representation, our experimental results indicate that it is not a good pooling operator for representing the appearance model in visual tracking. The reason is that max pooling increases the invariance while losing much discriminative elements. Comparison experiments in object recognition also indicate that, when an appropriate rectification layer is adopted, sum pooling slightly outperforms max pooling [42]. Therefore, in this paper, we choose to use sum pooling rather than max pooling. However, sum pooling over the entire image has a significant drawback — the spatial relationships of the local codes will be lost. Since the spatial relationships are important for visual tracking, we propose to use spatial pyramid pooling proposed in [67], which effectively preserves the spatial relationships and effectively increases the global discriminative capability with sufficient local robustness against appearance variations.

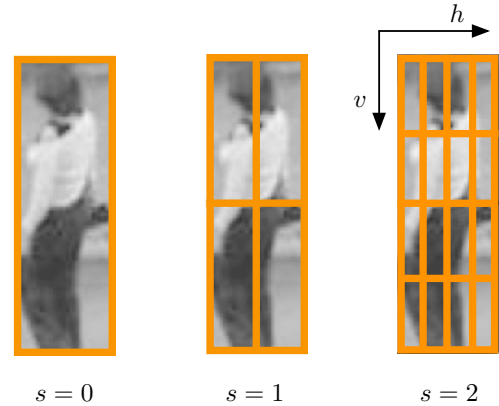


Fig. 2. The spatial pyramid structure with three scales.

In particular, we evenly partition the input image into  $2^s \times 2^s$  sub-images in three different scales  $s = \{0, 1, 2\}$ . See an example shown in Fig. 2. Let  $v$  and  $h$  be the indices of the sub-images along the vertical and horizontal directions, respectively. The  $(v, h)$ -th sub-image in the scale  $s$  is denoted as  $\mathbf{I}_{s,v,h}$  where  $v = 1, 2, \dots, 2^s$  and  $h = 1, 2, \dots, 2^s$ . Let  $\mathbf{N}_{s,v,h} = [\mathbf{n}_{s,v,h,1}, \mathbf{n}_{s,v,h,2}, \dots, \mathbf{n}_{s,v,h,L}] \in \mathbb{R}^{K \times L}$  be the normalization layer outputs of  $L$  image patches sampled in the sub-image  $\mathbf{I}_{s,v,h}$ . Because each patch in the sub-image is encoded by all the basis functions, the distribution of responses of all the basis functions can be used as features to describe the sub-image. Taking this into account, the feature representation

---

**Algorithm 2:** The proposed tracking algorithm using a novel appearance modeling technique.

---

**Input :** Image sequence  $\{\mathbf{I}_t\}_{t=1:T}$ , initial target state  $\mathbf{z}_0$ , target templates  $\{\hat{\mathbf{I}}_0^j\}_{j=1:J}$ , parameters  $N$  and  $\Sigma$

**Output:** The tracking results  $\{\hat{\mathbf{y}}_0^j\}_{j=1:J}$

- 1 Compute appearance representations  $\{\hat{\mathbf{y}}_0^j\}_{j=1:J}$  for template images  $\{\hat{\mathbf{I}}_0^j\}_{j=1:J}$  using Algorithm 1;
- 2 **for**  $t = 1, 2, \dots$  **do**
- 3     Sample a set of  $N$  particle  $\{\mathbf{z}_1^i\}_{i=1:N}$  from  $\mathcal{N}(\mathbf{z}_{t-1}, \Sigma)$ ;
- 4     Determine images  $\{\mathbf{I}_t^i\}_{i=1:N}$  from target candidates parameterized using  $\{\mathbf{z}_t^i\}_{i=1:N}$ ;
- 5     Compute appearance representations  $\{\mathbf{y}_t^i\}_{i=1:N}$  for  $\{\mathbf{I}_t^i\}_{i=1:N}$  using Algorithm 1;
- 6     Compute particle weights  $\{w_t^i\}_{i=1:N}$  using Eq. (14);
- 7     Obtain the tracking result at time  $t$ ,  $\mathbf{z}_t^{i^*}$  using Eq. (15);
- 8     Update target templates using the procedure described in Section IV-C;
- 9     Resample  $N$  particles  $\{\mathbf{z}_t^i\}_{i=1:N}$  with replacement from the current particle set according to probabilities  $\{w_t^i\}_{i=1:N}$ ;
- 10 **end**

---

$\mathbf{y}_{s,v,h}$  of the sub-image  $\mathbf{I}_{s,v,h}$  can be obtained by sum pooling along each dimension over all the image patches in the sub-image, which calculates the activation frequencies of basis functions when encoding the image patches in the sub-image. For example, the  $k$ -th element of  $\mathbf{y}_{s,v,h}$  is obtained by

$$\mathbf{y}_{s,v,h}(k) = \mathcal{P}(\mathbf{N}_{s,v,h}(k)) = \sum_{l=1}^L \mathbf{n}_{s,v,h,l}(k) \quad (8)$$

where  $\mathbf{N}(k)$  denotes the  $k$ -th row of the matrix  $\mathbf{N}$  and  $\mathbf{n}(k)$  the  $k$ -th element of the vector  $\mathbf{n}$ . After pooling along all the dimensions, we then normalize  $\mathbf{y}_{s,v,h}$  to make the sum of its all elements to be one. The feature set  $\{\mathbf{y}_{s,v,h}|s = 0, 1, 2, v = 1, 2, \dots, 2^s, h = 1, 2, \dots, 2^s\}$ , computed from all the sub-images, are used as the final appearance representation of the input image.

Please note that the multi-scale pooling is implemented for each input image to compute its appearance representation. Here the input image can be either a target template image (as shown in Fig. 2) or a target candidate image (e.g., an image with the same size as the target image but which may contain some background regions). The number of all the sub-images in all the three scales is 21.

#### IV. PROPOSED TRACKING ALGORITHM

Based on the proposed appearance model presented above, in this section, we introduce the proposed tracking algorithm using a multi-scale pyramid matching scheme within a standard particle filter framework. The detailed algorithm for tracking is outlined in Algorithm 2.

##### A. Particle filter framework

Particle filtering [68] is a popular computation method to recursively approximate the posterior distribution of the state variables characterizing a dynamic system. It consists of two stages: prediction and update. Let  $\mathbf{z}_t$  and  $\mathbf{I}_t$  be the state variable and the observation at time  $t$ , respectively. The posterior distribution of  $\mathbf{z}_t$  given all the available observations  $\mathbf{I}_{1:t-1} = \{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_{t-1}\}$  up to time  $t-1$  can be predicated using the state transition model  $p(\mathbf{z}_t|\mathbf{z}_{t-1})$  as:

$$p(\mathbf{z}_t|\mathbf{I}_{1:t-1}) = \int p(\mathbf{z}_t|\mathbf{z}_{t-1})p(\mathbf{z}_{t-1}|\mathbf{I}_{1:t-1})d\mathbf{z}_{t-1} \quad (9)$$

At time  $t$ , the observation  $\mathbf{I}_t$  is available, the posterior distribution of  $\mathbf{z}_t$  is updated using the Bayes' rule as

$$p(\mathbf{z}_t|\mathbf{I}_{1:t}) = \frac{p(\mathbf{I}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{I}_{1:t-1})}{p(\mathbf{I}_t|\mathbf{I}_{1:t-1})} \quad (10)$$

Using a sequential importance sampling technique, we can approximate the posterior distribution  $p(\mathbf{z}_t|\mathbf{I}_{1:t})$  by a set of  $N$  weighted samples (also called particles)  $\{\mathbf{z}_t^i, w_t^i\}_{i=1,\dots,N}$ , where  $w_t^i$  are the importance weights of particles  $\mathbf{z}_t^i$ . Let  $q(\mathbf{z}_t|\mathbf{I}_{1:t}, \mathbf{z}_{1:t-1})$  be the importance distribution from which the particles are drawn, the importance weights  $w_t^i$  are updated as:

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{I}_t|\mathbf{z}_t^i)p(\mathbf{z}_t^i|\mathbf{z}_{t-1}^i)}{q(\mathbf{z}_t|\mathbf{I}_{1:t}, \mathbf{z}_{1:t-1})} \quad (11)$$

To avoid the degeneracy case where the weights of some particles possibly keep increasing, particles are resampled according to their importance weights so as to generate a set of equally weighted particles. In the case of bootstrap filtering [68], the state transition distribution is chosen as the importance distribution  $q(\mathbf{z}_t|\mathbf{I}_{1:t}, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t|\mathbf{z}_{t-1}) \sim \mathcal{N}(\mathbf{0}, \Sigma)$  where  $\Sigma$  is a diagonal matrix, the weights are updated as the observation likelihood  $w_t^i = p(\mathbf{I}_t|\mathbf{z}_t^i)$ .

Particle filtering has found its applications in contour tracking [69]. Perez et al. [70] used particle filtering for tracking a target parameterized by a rectangle region. The key step of particle filtering for visual tracking is to compute the weight for each particle using the observation likelihood. In practice, the observation likelihood  $p(\mathbf{I}_t|\mathbf{z}_t^i)$  is computed as the similarity between the target template image and the target candidate image parameterized by the particle  $\mathbf{z}_t^i$ . In the next subsection, we present how to combine our proposed appearance model with a multi-scale pyramid matching to assign an appropriate weight to each particle.

##### B. Particle weighting based on the proposed appearance model

To compute the weight  $w_i$  for the  $i$ -th particle  $\mathbf{z}_i$ , we first crop a candidate image  $\mathbf{I}_i$  parameterized by  $\mathbf{z}_i$  from the current image<sup>1</sup>. Using the proposed appearance model, we can compute a set of feature descriptors  $\{\mathbf{y}_{s,v,h}^i|s = 0, 1, 2, v = 1, 2, \dots, 2^s, h = 1, 2, \dots, 2^s\}$  from the candidate image  $\mathbf{I}_i$ . In addition, we maintain a set of  $J$  target templates  $\{\hat{\mathbf{I}}_1, \hat{\mathbf{I}}_2, \dots, \hat{\mathbf{I}}_J\}$ . The feature descriptors of the  $j$ -th

<sup>1</sup>the time index is omitted for readability

target template are  $\{\hat{\mathbf{y}}_{s,v,h}^j | s = 0, 1, 2, v = 1, 2, \dots, 2^2, h = 1, 2, \dots, 2^s\}$ . The distance between the  $j$ -th template  $\hat{\mathbf{I}}_j$  and the candidate  $\mathbf{I}_i$  is computed by summing the pairwise distance between their corresponding descriptors, shown as follows:

$$\rho(\mathbf{I}_i, \hat{\mathbf{I}}_j) = \sum_{s=0}^2 \sum_{v=1}^{2^s} \sum_{h=1}^{2^s} \sum_{k=1}^K \frac{(\mathbf{y}_{s,v,h}^i(k) - \hat{\mathbf{y}}_{s,v,h}^j(k))^2}{\mathbf{y}_{s,v,h}^i(k) + \hat{\mathbf{y}}_{s,v,h}^j(k)} \quad (12)$$

Based on the above distance, we further define the similarity between the  $j$ -th template  $\hat{\mathbf{I}}_j$  and the candidate  $\mathbf{I}_i$  as

$$\eta(\mathbf{I}_i, \hat{\mathbf{I}}_j) = \mathcal{D}(\rho(\mathbf{I}_i, \hat{\mathbf{I}}_j); 0, \sigma^2) \quad (13)$$

where  $\mathcal{D}(x; \mu, \sigma^2)$  is the Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ . We fix  $\mu = 0$  and  $\sigma^2$  to 15 in all the experiments reported in this paper.

We use the weighted sum of the similarities between the candidate  $\mathbf{I}_i$  and all the target templates  $\{\hat{\mathbf{I}}_1, \hat{\mathbf{I}}_2, \dots, \hat{\mathbf{I}}_J\}$  as the weight of the  $i$ -th particle:

$$w_i = \sum_{j=1}^J \pi_j \eta(\mathbf{I}_i, \hat{\mathbf{I}}_j) \quad (14)$$

where  $\pi_j$  is the weight associated to  $j$ -th target template image. All the target templates are initialized to have the same weights and then the weights are updated using the procedure that will be introduced in subsection IV-C.

The tracking result in the current frame is the particle with the largest weight with an index computed as

$$i^* = \arg \max_{i=1, \dots, N} w_i \quad (15)$$

Please note that discriminative approaches such as SVM or structured SVM can be used for visual tracking with the proposed appearance model. However, in this paper we still use distance matching (Eq. 12) due to its simplicity and efficiency.

### C. Online multiple template update

In order to capture the target's appearance variations during tracking, we maintain a set of target templates and update them online. For the purpose of dynamical updating, we assign a weight to each template, which reflects how much the template is similar to the tracked target. After the target has been tracked, the tracking result will be added to the template set so that we capture the latest target appearance. However, we do not know if the tracking result is correct. Therefore, to avoid the accumulation of errors, we only include the recent tracking results that are similar enough to the corresponding templates for template updating. The index used to indicate this similarity is represented by  $j^* = \arg \max_j \eta(\mathbf{I}_{i^*}, \hat{\mathbf{I}}_j)$ . We set a threshold  $\tau$  so that if the cosine angle between  $\mathbf{I}_{i^*}$  and  $\hat{\mathbf{I}}_{j^*}$  is less than the threshold, we then replace the  $J$ -th template  $\hat{\mathbf{I}}_J$  using the available tracking result  $\mathbf{I}_{i^*}$ . The weight  $\pi_J$  of the  $J$ -th template is assigned to be the median of the previous weights of all the templates before the template updating starts. The weights of the remaining templates are updated as  $\pi_j = \eta(\mathbf{I}_{i^*}, \hat{\mathbf{I}}_j)$  where  $j = 1, 2, \dots, J-1$ . After the updating, the weights of all the templates are then normalized

to have the sum as one and the templates are finally sorted in a descending order according to their normalized weights.

## V. EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed method. We introduce the implementation details and the evaluation protocol in sections V-A and V-B, respectively, including the parameter setup, baseline trackers, test sequences and evaluation criteria. In section V-C, we evaluate the proposed method when different layer architectures are used. The comparison results against the other baseline trackers are presented in section V-D.

### A. Implementation details

For each sequence, the four corner points of the target is manually labeled in the first frame. Similar to [71], the state variable  $\mathbf{z}_t$  consists of six affine transformation parameters. We set the variance matrix of the importance distribution  $\Sigma = [0.005, 0.0005, 0.0005, 0.005, 3, 3]^T$  for all the sequences. By applying affine transformation using  $\mathbf{z}_0$  as the initial parameters, we define the target template image as the first image frame and then duplicate it ten times to form the target template set. To learn the dictionary, we obtain a foreground patch set by densely extracting  $8 \times 8$  local patches from the target template images with 4 pixels as the step length. The background patch set is obtained by sampling  $8 \times 8$  patches around the target region. For all the experiments, we manually set the number of particles to be  $N = 600$ , the number of basis functions of the dictionaries  $K_f = K_b = 64$ , the regularization parameter  $\lambda = 0.01$  to achieve a tradeoff between the tracking accuracy and speed. The proposed algorithm is implemented in MATLAB and runs around 4 frames per second on a PC with a 2.4 GHz Intel Core i5 processor and a 4GB memory.

### B. Experiment Protocols

Recently, a large scale benchmark library<sup>2</sup> for visual tracking was built by Wu *et al.* [1] and further extended [54], which contains the source codes of most state-of-the-art trackers and a large number of annotated test sequences. Most importantly, they propose robust evaluation metrics, which are used to reliably measure the performance of a tracker on different challenging scenarios. In this work, we follow the evaluation protocols reported in the literature. In particular, we choose all the 100 test sequences from their dataset including different tracking challenges such as illumination variation, low resolution, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view and background clutter.

To make the comparisons fair, we deliberately chose two state of the art sparse coding based trackers including the L1 tracker using accelerated proximal gradient (L1APG) [72] and the online robust non-negative dictionary learning (ON-NDL) [73]. We also selected five latest trackers including the deep learning based tracker (DLT) [74], Complex Cell based Tracker (CCT) [52], the self-correction ensemble-based

<sup>2</sup><http://visual-tracking.net>



tracking (SC-EBT) [75], the transfer learning based visual tracking with Gaussian process regression (TGPR) [76], and the Kernelized Correlation Filters based tracker (KCF) [6] for comparisons. For these trackers, we run the publicly available source codes on the benchmark in order to obtain their results. Please note that the results reported here may be of slight difference from the results reported in their original papers due to the utilization of parameters.

Two frame based metrics widely used in tracking performance evaluation are: 1) *center location error*, which is defined as the Euclidean distance between the central location of the tracked target and the manually labeled ground-truthed position; 2) *bounding box overlap* which is the ratio of the areas of the intersection and the union of the bounding box indicating the tracked subject and the ground-truthed bounding box. To measure the overall performance of a tracker on a test sequence, *success rate* and *precision score* are adopted. The former is computed as the percentage of image frames which have a bounding box overlap larger than a given threshold. The later is the percentage of image frames which have a central position error less than a given threshold. In each case, when multiple thresholds are used, a curve is drawn to show how success rates or precision scores are affected by different thresholds. These curves are namely **Success plot** and **Precious plot**, respectively. In practical evaluations, we average the curves of a tracker over all the sequences which have the same challenge and show a curve for each challenge item rather than a test sequence. In addition, we use the area under curve (AUC) of the success plot to quantitatively measure the overall performance of a tracker on a challenge item.

The conventional way to evaluate trackers is to run a tracker throughout a test sequence with an initialization from the ground-truthed position in the first frame. However, we found that the initialization usually affects the performance of a tracker significantly. Therefore, it is necessary to test how robust a tracker is against different initialization states. In [1], Wu *et al.* proposed two ways to analyze a tracker's robustness against initialization: temporal robustness evaluation (TRE) that perturbs the initialization by starting a tracker at different frames and spatial robustness evaluation (SRE) that perturbs the initialization spatially by starting a tracker at different bounding boxes. In this work, we adopt the SRE for all the comparison shown in this paper.

### C. Comparison with different layer architectures

We compare different layer architectures on eight test sequences. First of all, we study whether or not the discriminative sparse coding in the coding layer and the multiple scale pyramid representation in the pooling layer help to improve the tracking accuracy. To find out the benefits of using different pyramid scales, we design seven experiments with multiple pyramid scale  $s = \{0\}$ ,  $s = \{1\}$ ,  $s = \{2\}$ ,  $s = \{0, 1\}$ ,  $s = \{0, 2\}$  and  $s = \{0, 1, 2\}$ . For each scale, to evaluate the benefits of using discriminative sparse coding, we compare the proposed tracker using the discriminative basis functions  $\mathbf{V}_d$  and the one using the generic basis functions

that are learned using local patches extracted from the target image via Eq. 3. The AUC of these experiments are shown in Table I, where for each scale, the left column indicates the results when the discriminative basis functions are used and the right column refers to the results when the generic basis functions are used. It clearly shows that for most of the test sequences, pyramid scale  $s = \{2\}$  leads to the highest tracking success rates. In the meantime,  $s = \{0\}$  (corresponding to no pyramid structure) has the worst results. These results justify that pyramid representation especially with a large scale helps to achieve good visual tracking accuracy. From this table, we also observe that using the discriminative basis functions in the coding layer brings us better results than only using general basis functions.

We further investigate whether or not all the five layers are necessary for the proposed appearance model. We choose the architecture with all the five layers as the basic architecture and then remove or replace some of the operators used in the basic architecture to form different architectures. For example, if the whitening layer or the normalization layer is removed from the basic architecture, we call it *basic- $\mathcal{W}_{no}$*  or *basic- $\mathcal{N}_{no}$* , respectively. If we replace the max operator in the rectification layer with an absolute or square operator, we call them *basic- $\mathcal{R}_{abs}$*  or *basic- $\mathcal{R}_{sq}$* . If we replace the sum operator in the pooling layer with a max operator, we call it *basic- $\mathcal{P}_{max}$* . Table II shows the tracking success rates of such experiments based on the first eight test sequences, from which we can see that the basic architecture achieves the best accuracy on all the test sequences. When the whitening layer is removed, the tracking success rates on all the sequences witness significant drops. The reason is that the whitening operator helps speedup the convergence of the dictionary learning in the coding layer and if it is removed, then the system accuracy degrades. From the comparison of using different rectification operators, we observe that the max operator outperforms both the absolute and the square operators. As explained in the previous section, the max operator increases the sparsity of the codes obtained in the coding layer and hence reduces the influence of the background. Finally, we also see that the sum pooling based method outperforms the max pooling based.

### D. Comparison against different methods

We compare the proposed method with seven trackers on all the 100 sequences used in [54]. Figs. 3 and 4 show the success plots and precious plots for all the involved trackers over all the test sequences containing a specific challenge (e.g., fast motion, occlusion) as well as the overall success plots and precious plots over all the test sequences. Figs. 3(a), 3(b), 3(c), 3(d), 4(e), 4(f), 3(h), 3(i), and 3(j) show the success plots of the studied trackers for the challenges of fast motion, background clutter, motion blur, deformations, illumination variations, in-plane rotation, occlusions, out-plane rotation and out of view, respectively. Our tracker significantly outperforms the other trackers with different thresholds.

Low resolution, fast motion and scale variation are particularly challenging for visual tracking. As shown in Figs. 3(g), 3(a) and 3(k) (and also Figs. 4(g) and 4(a)), our tracker

TABLE I

THE TRACKING SUCCESS RATES ON EIGHT TEST SEQUENCES BY THE PROPOSED TRACKER USING DIFFERENT PYRAMID SCALES AND DISCRIMINATIVE SPARSE CODING OR GENERIC SPARSE CODING. FOR EACH SEQUENCE, THE BEST RESULT IS HIGHLIGHTED IN RED COLOR AND HEREFTER.

	$s = \{0\}$		$s = \{1\}$		$s = \{2\}$		$s = \{0, 1\}$		$s = \{0, 2\}$		$s = \{1, 2\}$		$s = \{0, 1, 2\}$	
<i>car4</i>	0.37	0.19	0.38	0.21	0.57	0.39	0.36	0.21	0.80	0.61	<b>0.89</b>	0.68	0.57	0.37
<i>david_outdoor</i>	0.28	0.21	0.32	0.25	<b>0.95</b>	0.85	0.75	0.59	0.69	0.59	0.76	0.57	0.44	0.34
<i>david_indoor</i>	0.36	0.19	0.33	0.24	<b>1.00</b>	0.79	0.36	0.21	0.77	0.61	0.85	0.64	0.69	0.44
<i>sylv</i>	0.86	0.72	0.56	0.47	0.62	0.52	<b>1.00</b>	0.77	0.79	0.65	0.98	0.72	0.96	0.79
<i>lemming</i>	0.18	0.11	0.18	0.12	<b>0.63</b>	0.40	0.19	0.13	0.35	0.21	0.45	0.35	0.57	0.41
<i>box</i>	0.71	0.62	0.86	0.77	<b>0.95</b>	0.85	0.92	0.79	0.63	0.50	0.69	0.58	0.92	0.72
<i>basketball</i>	0.21	0.14	0.38	0.21	<b>0.76</b>	0.58	0.42	0.31	0.55	0.46	0.52	0.37	0.47	0.35
<i>faceocc2</i>	0.82	0.65	0.75	0.59	0.70	0.53	0.83	0.69	<b>1.00</b>	0.79	0.84	0.75	0.83	0.62

TABLE II

THE TRACKING SUCCESS RATES ON THE EIGHT TEST SEQUENCES BY THE PROPOSED TRACKER USING DIFFERENT LAYER ARCHITECTURES.

	<i>basic</i>	<i>basic-<math>\mathcal{W}_{no}</math></i>	<i>basic-<math>\mathcal{R}_{abs}</math></i>	<i>basic-<math>\mathcal{R}_{sq}</math></i>	<i>basic-<math>\mathcal{N}_{no}</math></i>	<i>basic-<math>\mathcal{P}_{max}</math></i>
<i>car4</i>	<b>0.89</b>	0.60	0.73	0.71	0.80	0.81
<i>david_outdoor</i>	<b>0.95</b>	0.69	0.75	0.80	0.85	0.93
<i>david_indoor</i>	<b>1.00</b>	0.77	0.84	0.81	0.90	0.94
<i>sylv</i>	<b>1.00</b>	0.79	0.86	0.81	0.90	0.92
<i>lemming</i>	<b>0.63</b>	0.46	0.52	0.51	0.55	0.51
<i>box</i>	<b>0.95</b>	0.77	0.86	0.81	0.89	0.87
<i>basketball</i>	<b>0.76</b>	0.60	0.66	0.70	0.72	0.68
<i>faceocc2</i>	<b>1.00</b>	0.72	0.90	0.86	0.92	0.94

is slightly worse than TGPR, which may be caused by our dictionary learning methods on sampled patches with size  $8 \times 8$ . For low resolution or blurred videos caused by fast motion, the sampled patches will contain few information, which may cause the learned dictionaries less discriminative.

In Table III, we provide the AUC values for all the involved trackers per each challenging scenario and the overall performance on the whole dataset. As we can see, our tracker achieved the best AUC in nine out of eleven challenges. In terms of the overall AUC, our tracker outperforms all the other trackers. Precious scores of the studied trackers are shown in Table IV, from which we can see that our tracker achieved the best AUC in ten out of eleven challenges and the best overall AUC.

1) *Comparison on tracking speed*: To compare the computational efficiency, we compute the time of running the compared trackers on the *car4* test sequence and reported the tracking speeds (frames / second) in Table V. From this table, we can see that our tracker can achieve about 4 frames per second. Although it is not real time, it is still faster than the ONNDL tracker and the SC-EBT tracker. With code optimisation in C++ or running on GPU, our tracker has potential to achieve real-time tracking. To see the computational cost of each step of our tracker, we also reported the running time of each step of our tracker on the *car4* sequence. From Table VI, we can see that the most computational expensive step is the coding step.

2) *Discussion*: A number of biologically inspired appearance models that simulate the hierarchical pathway in visual cortex have been extensively studied in the last two decades, and these existing appearance models were mainly developed for object recognition. The challenge in object recognition is to recognize the same class of objects in the presence of different object transformation such as translation, scale, and pose/view/illumination changes. To achieve the tasks, the

appearance models for object recognition are required to be invariant to translation, scale, view and illumination changes. Visual tracking indeed is a recognition problem but it has its own characteristics. For example, in a standard particle filter based tracker, the appearance model has not considered scale changes and people can take advantage of the temporal consistence constraint to identify the size changes of the target. This makes appearance models used for visual tracking different from those used for object recognition. In particular, visual tracking is a binary classification problem. In this work, motivated by the success of using the biologically inspired appearance models for object recognition, we developed a biologically inspired appearance model for visual tracking with an emphasis on both discriminativity and appearance invariance. We take advantage of the established hierarchical architecture in visual cortex while introducing the discriminative sparse coding and spatial pyramid representation to the tracking system. Although the proposed appearance model is evaluated in the context of visual tracking, the proposed framework is easy to be extended to other multiple-class classification tasks with well-aligned samples.

## VI. CONCLUSION AND FUTURE WORK

In this paper, motivated by the success of the hierarchical organization of the primary visual cortex in object recognition, we have proposed a biologically inspired appearance model for robust visual tracking. The proposed appearance model consists of five layers, simulating the visual information processing pathway from retina to complex cells of the primary visual cortex. Different from the existing biologically inspired appearance models reported in the literature, which over-emphasize the invariance to translation, scale and view changes while ignoring the discriminative ability, the proposed model made a good compromise between the discriminative

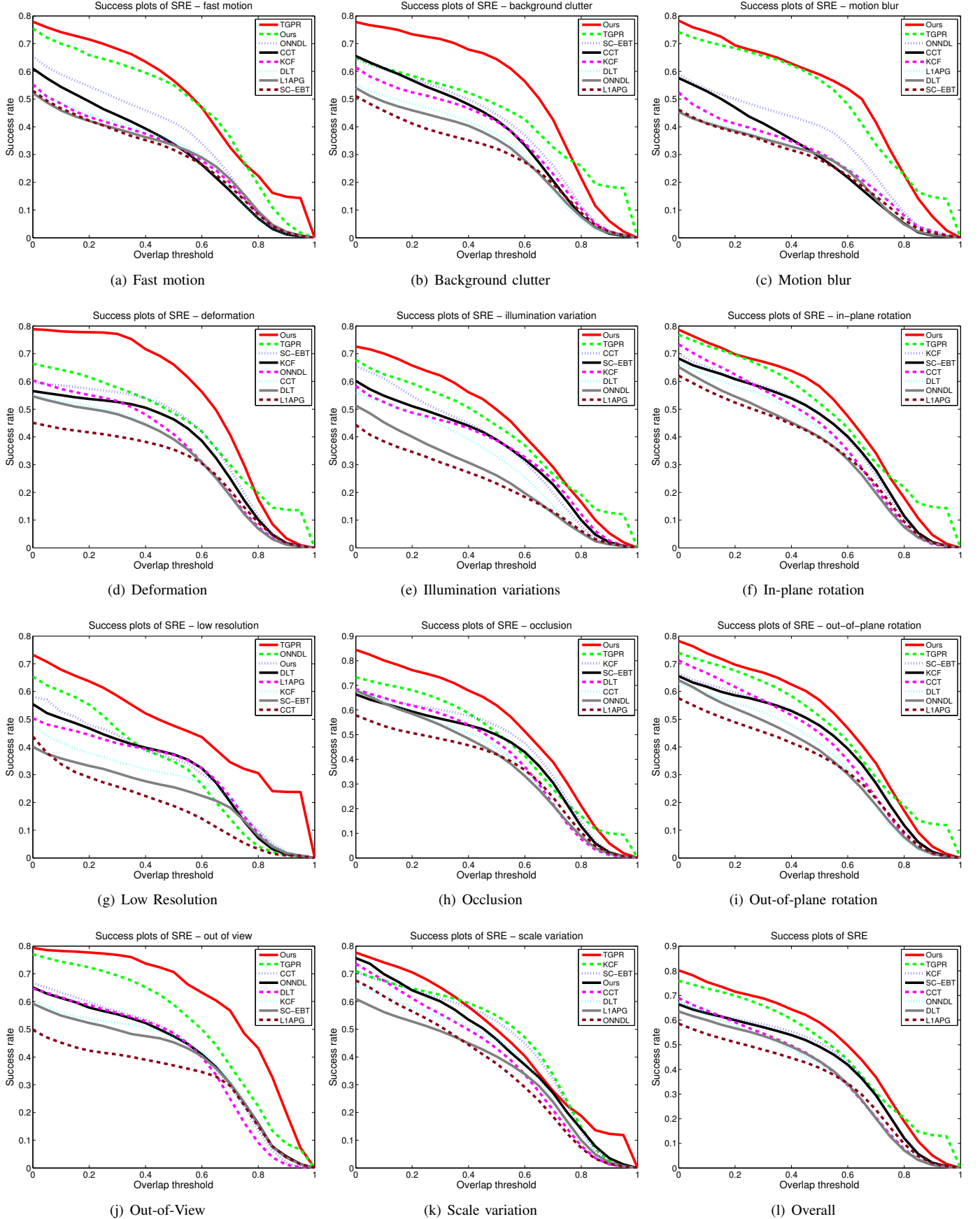


Fig. 3. Success plots for the challenges considered in this work.

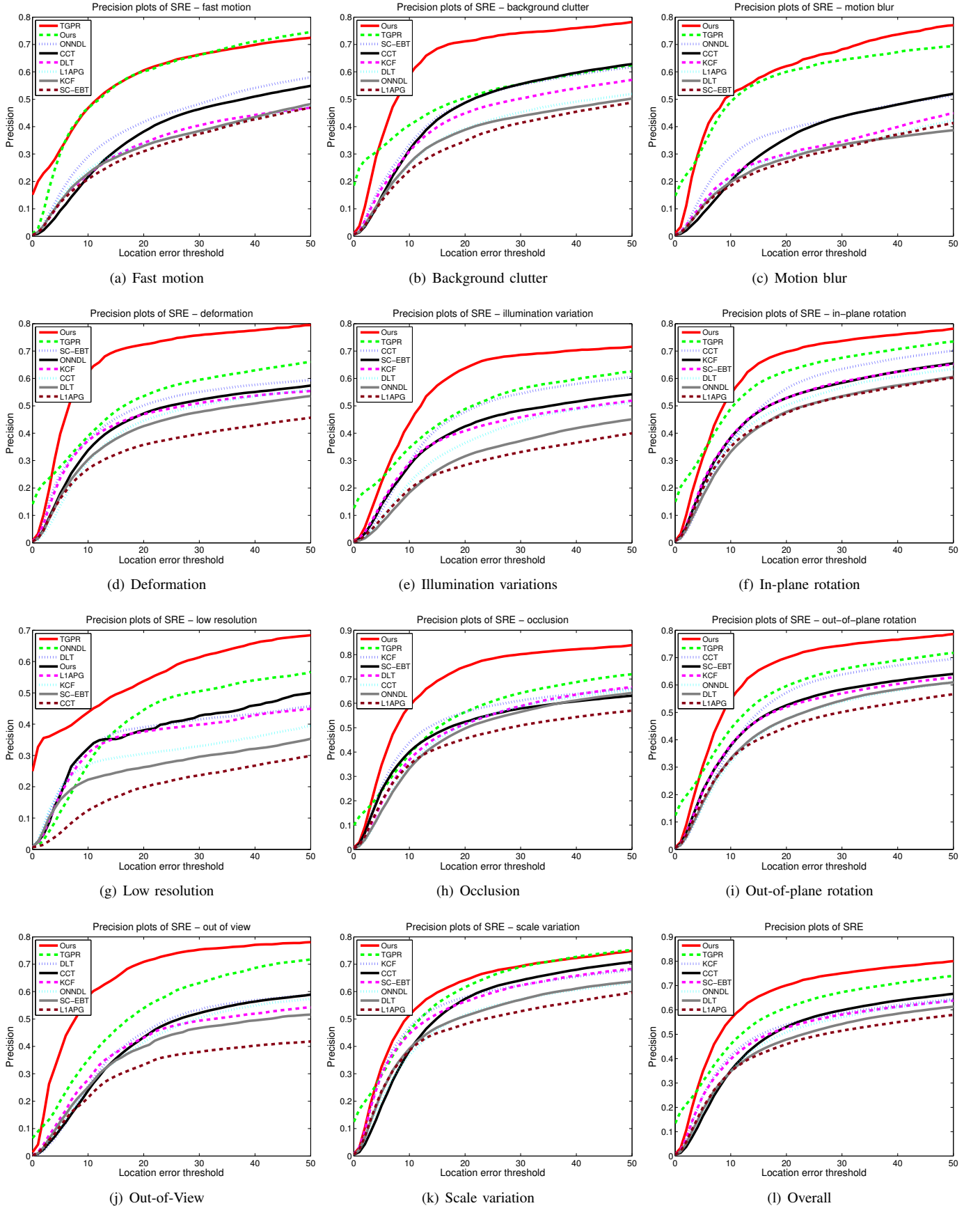


Fig. 4. Precision plots for the challenges considered in this work.



	Ours	L1APG	ONNDL	DLT	CCT	SC-EBT	KCF	TGPR
Occlusion	0.514	0.339	0.363	0.386	0.372	0.400	0.424	0.443
Illumination variation	0.428	0.215	0.242	0.292	0.341	0.331	0.328	0.400
Scale variation	0.412	0.342	0.341	0.376	0.377	0.428	0.438	0.454
Background clutter	0.512	0.274	0.295	0.304	0.356	0.373	0.345	0.425
Deformation	0.522	0.280	0.341	0.316	0.320	0.385	0.358	0.422
Fast motion	0.454	0.283	0.350	0.288	0.303	0.275	0.288	0.487
Motion blur	0.492	0.255	0.328	0.244	0.275	0.242	0.267	0.477
In-plane rotation	0.472	0.339	0.342	0.363	0.385	0.397	0.398	0.467
Out-of-plane rotation	0.467	0.316	0.334	0.345	0.382	0.390	0.387	0.448
Out of view	0.589	0.309	0.394	0.382	0.398	0.365	0.380	0.492
Low resolution	0.312	0.297	0.314	0.0.304	0.0.179	0.222	0.260	0.458
Overall	0.485	0.335	0.360	0.355	0.366	0.398	0.3405	0.466

TABLE III

AUC COMPARISON OF THE STUDIED TRACKERS PER EACH CHALLENGING SCENARIO. NOTE THE LAST ROW IS NOT SIMPLY THE AVERAGE OF THE OVERALL ROWS.

	Ours	L1APG	ONNDL	DLT	CCT	SC-EBT	KCF	TGPR
Occlusion	0.750	0.454	0.496	0.515	0.513	0.524	0.563	0.564
Illumination variation	0.637	0.284	0.316	0.364	0.478	0.425	0.409	0.488
Scale variation	0.642	0.481	0.515	0.511	0.573	0.563	0.580	0.616
Background clutter	0.711	0.348	0.389	0.393	0.487	0.488	0.449	0.505
Deformation	0.724	0.358	0.472	0.426	0.446	0.502	0.469	0.535
Fast motion	0.602	0.334	0.418	0.0.341	0.382	0.310	0.329	0.605
Motion blur	0.621	0.293	0.391	0.284	0.356	0.271	0.301	0.600
In-plane rotation	0.697	0.473	0.478	0.502	0.563	0.529	0.529	0.627
Out-of-plane rotation	0.700	0.446	0.476	0.476	0.573	0.526	0.520	0.596
Out of view	0.709	0.333	0.416	0.450	0.434	0.399	0.431	0.539
Low resolution	0.381	0.376	0.445	0.391	0.198	0.263	0.306	0.538
Overall	0.701	0.459	0.508	0.478	0.531	0.527	0.539	0.608

TABLE IV

PRECIOUS SCORES FOR THE STUDIED TRACKERS PER EACH CHALLENGING SCENARIO. NOTE THAT THE PRECIOUS SCORE IS COMPUTED FOR THE THRESHOLD VALUE OF 20.

	Ours	L1APG	ONNDL	DLT	CCT	SC-EBT	KCF	TGPR
Tracking speed	4	25	0.5	5	9	0.8	120	0.59

TABLE V

AVERAGE TRACKING SPEED (FRAMES PER SECOND) TESTED ON THE *car4* SEQUENCE.

	Whitening	Coding	Rectification	Normalization	Pooling	distance computation
Running time (in second)	20.83	87.56	12.16	18.89	15.92	23.26

TABLE VI

RUNNING TIME (IN SECONDS) OF EACH STEP TESTED ON THE *car4* SEQUENCE.

ability and its robustness against appearance variations. Experimental results show that the proposed model is reasonable and achieved promising performance. Comparison results against several state-of-the-art tracking methods also validated the effectiveness of the proposed method.

The proposed appearance modelling method was inspired by biological findings in object recognition of still images and achieved desired tracking accuracy for video applications. Our work disregarded the dynamic nature of cortical computation and the representation of cortical information over time. In the future work, we will develop learning tools to model the temporal relationship between cortical responses, which may further improve the tracking accuracy.

## VII. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (Nos. 61300111 and 61472103)

and Key Program (No. 61133003); in part by the Australian Research Council Projects DP-140102164, FT-130101457, and LE140100061. H. Zhou is also supported by UK EPSRC under Grants EP/G034303/1, EP/N508664/1 and EP/N011074/1.

## REFERENCES

- [1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418, 2013.
- [2] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 46, no. 7, pp. 1772–1788, 2013.
- [3] H. Zhou, M. Fei, A. Sadka, Y. Zhang, and X. Li, "Adaptive fusion of particle filtering and spatio-temporal motion energy for human tracking," *Pattern Recognition*, vol. 47, no. 11, pp. 3552–3567, 2014.
- [4] S. Zhang, H. Zhou, H. Yao, Y. Zhang, K. Wang, and J. Zhang, "Adaptive normalhedge for robust visual tracking," *Signal Processing*, vol. 110, pp. 132–142, 2015.
- [5] X. Yang, M. Wang, and D. Tao, "Robust visual tracking via multi-graph ranking," *Neurocomputing*, vol. 159, pp. 35–43, 2015.

- [6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015.
- [7] L. Zhao, X. Gao, D. Tao, and X. Li, "Tracking human pose using max-margin markov models," *IEEE Transaction on Image Processing*, vol. 24, no. 12, pp. 5274–5287, 2015.
- [8] S. Zhang, H. Zhou, F. Jiang, and X. Li, "Robust visual tracking using structurally random projection and weighted least squares," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1749–1760, 2015.
- [9] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1818–1828, 2015.
- [10] L. Zhao, X. Gao, D. Tao, and X. Li, "Learning a tracking and estimation integrated graphical model for human pose tracking," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 12, pp. 3176–3186, 2015.
- [11] X. Mei, Z. Hong, D. V. Prokhorov, and D. Tao, "Robust multitask multiview tracking in videos," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 11, pp. 2874–2890, 2015.
- [12] Z. Hong, Z. Chen, C. Wang, X. Mei, D. V. Prokhorov, and D. Tao, "MULTI-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 749–758, 2015.
- [13] D. Tao, L. Jin, Y. Wang, and X. Li, "Person reidentification by minimum classification error-based kiss metric learning," *IEEE Trans. Cybernetics*, vol. 45, no. 2, pp. 242–252, 2015.
- [14] J. Pan, X. Li, X. Li, and Y. Pang, "Incrementally detecting moving objects in video with sparsity and connectivity," *Cognitive Computation*, vol. 8, no. 3, pp. 420–428, 2016.
- [15] M. A. A. Aziz, J. Niu, X. Zhao, and X. Li, "Efficient and robust learning for sustainable and reacquisition-enabled hand tracking," *IEEE Trans. Cybernetics*, vol. 46, no. 4, pp. 945–958, 2016.
- [16] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.
- [17] J. Wen, X. Gao, Y. Yuan, D. Tao, and J. Li, "Incremental tensor biased discriminant analysis: A new color-based visual tracking method," *Neurocomputing*, vol. 73, pp. 827–839, 2010.
- [18] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983–990, 2009.
- [19] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1313–1320, 2011.
- [20] X. Jia, H. Lu, and M. Yang, "Visual tracking via adaptive structural local sparse appearance model," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1822–1829, 2012.
- [21] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Trans. Neur. Netw.*, vol. 21, no. 10, pp. 1610–1623, Oct. 2010.
- [22] X. Lu, Y. Yuan, and P. Yan, "Robust visual tracking with discriminative sparse learning," *Pattern Recognition*, vol. 46, no. 7, pp. 1762–1771, 2013.
- [23] J. Fang, Q. Wang, and Y. Yuan, "Part-based online tracking with geometry constraint and attention selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 854–864, 2013.
- [24] Q. Wang, J. Fang, and Y. Yuan, "Multi-cue based tracking," *Neurocomputing*, vol. 131, pp. 227–236, 2014.
- [25] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic, "Efficient online subspace learning with an indefinite kernel for visual tracking and recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1624–1636, 2012.
- [26] Z. Xiao, H. Lu, and D. Wang, "L2-RLS-based object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1301–1309, 2014.
- [27] F. Yang, H. Lu, and M.-H. Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 2, pp. 242–254, 2014.
- [28] —, "Robust superpixel tracking," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1639–1651, 2014.
- [29] X. Liu, D. Tao, M. Song, L. Zhang, J. Bu, and C. Chen, "Learning to track multiple targets," *IEEE Transactions on Neural Networks and Learning Systems*, no. 99, 2015, DOI:10.1109/TNNLS.2014.2333751.
- [30] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
- [31] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005.
- [33] R. Collins, Y. Liu, and M. Leordeanu, "On-line selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.
- [34] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 8, pp. 125–141, 2007.
- [35] C. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2010.
- [36] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
- [37] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [38] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.
- [39] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.
- [40] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 994–1000, 2005.
- [41] N. Pinto, D. Cox, and J. DiCarlo, "Why is real-world visual object recognition hard," *PLoS Computational Biology*, vol. 4, no. 1, pp. 25–45, 2008.
- [42] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" *Proceedings of International Conference on Computer Vision*, pp. 2146–2153, 2009.
- [43] F. Attneave, "Some informational aspects of visual perception," *Psychological Review*, vol. 61, no. 3, pp. 183–193, 1954.
- [44] H. Barlow, *Possible principles underlying the transformation of sensory messages*. MIT, 1961.
- [45] D. Perrett and M. Oram, "Neurophysiology of shape processing," *Image and Vision Computing*, vol. 11, no. 6, pp. 317–333, 1993.
- [46] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1794–1801, 2009.
- [47] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 524–531, 2005.
- [48] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.
- [49] A. Coates and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization," *Proceedings of the 28th International Conference on Machine Learning*, pp. 921–928, 2011.
- [50] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," *Proceedings of International Conference on Computer Vision*, pp. 263–270, 2011.
- [51] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," *Proceedings of Neural Information Processing Systems*, pp. 809–817, 2013.
- [52] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," *Proceedings of International Conference on Computer Vision*, pp. 1113–1120, 2013.
- [53] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.
- [54] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *TPAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [55] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Online object tracking: A benchmark," *Proceedings of International Conference on Computer Vision*, pp. 1113–1120, 2013.

- [56] S. Zhang, H. Yao, and S. Liu, "Robust visual tracking using feature-based visual attention," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1150–1153, 2010.
- [57] Q. Wang, F. Chen, W. Xu, and M. Yang, "Online discriminative object tracking with local sparse representation," *Proceedings of IEEE Workshop on the Applications of Computer Vision*, pp. 425–432, 2012.
- [58] W. Zhong, H. Lu, and M. Yang, "Robust object tracking via sparsity-based collaborative model," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, 2012.
- [59] R. Douglas, C. Koch, M. Mahowald, K. Martin, and H. Suarez, "Recurrent excitation in neocortical circuits," *Science*, vol. 269, no. 5226, pp. 981–985, 1995.
- [60] M. Carandini and D. Heeger, "Summation and division by neurons in visual cortex," *Science*, vol. 264, pp. 1333–1336, 1994.
- [61] A. Bell and T. Sejnowski, "The independent components of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.
- [62] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [63] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Sig. Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.
- [64] P. Bush and T. Sejnowski, *The cortical neuron*. Oxford University Press, 1995.
- [65] S. Lyu, "Dependency reduction with divisive normalization: justification and effectiveness," *Neural Computation*, vol. 23, no. 11, pp. 2942–2973, 2011.
- [66] J. Mutch and D. G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45–57, 2008.
- [67] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169 – 2178, 2006.
- [68] A. Doucet, N. de Freitas, and N. Gordon, "Sequential monte carlo methods in practice," *Springer-Verlag*, 2001.
- [69] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [70] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Proceedings of European Conference on Computer Vision*, pp. 661–675, 2002.
- [71] X. Mei, H. Ling, and D. Jacobs, "Sparse representation of cast shadows via L1-regularized least squares," *Proceedings of International Conference on Computer Vision*, pp. 583–590, 2009.
- [72] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1830–1837, 2012.
- [73] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," *Proceedings of International Conference on Computer Vision*, 2013.
- [74] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," *Proc. Advances in Neural Information Processing Systems*, pp. 809–817, 2013.
- [75] —, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," *Proc. Int. Conference on Machine Learning*, pp. 1107–1115, 2014.
- [76] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian process regression," *Proceedings of European Conference on Computer Vision*, pp. 188–203, 2014.